47653/PYI/V165

# REVERSE DATA DE-SKEW METHOD AND SYSTEM

## CROSS-REFERENCE TO RELATED APPLICATIONS

This application claims the benefit of U.S. provisional application numbers 60/259,968 filed December 30, 2000, 60/260,079 filed January 4, 2001, 60/260,628 filed January 8, 2001, 60/261,868, filed January 10, 2001, 60/272,635, filed February 28, 2001, and 60/273,763, filed March 5, 2001 which are hereby incorporated by reference as if set forth in full herein.

## BACKGROUND OF THE INVENTION

The present invention relates generally to parallel data alignment, and more particularly to synchronization of high speed parallel data transmissions.

The capabilities of information processing systems are constantly expanding. Such systems are increasingly called upon to process large amounts of information very quickly. The ability of information processing systems to act on information is dependent on the rate at which the system may receive information and the speed at which the system can process that information. In order to receive information more quickly, the systems are often provided information on parallel data lines. The information provided on the parallel data lines is generally associated together to form blocks of information. The use of parallel data lines allows a system to receive multiple pieces of information at any given moment.

A problem with parallel data lines is that transmission times across the data lines may vary, or skew, due to line lengths, process variations, aging, and environmental conditions. If the data transmission times are sufficiently different, then the information processing system may not group pieces of information received on the data lines in the proper format. The increased rate at which information processing systems process information also results in a decreased tolerance of variation in data transmission time. Thus, if information processing speeds increased by a factor of 10, such has occurred in the last several years, the allowable variation in transmission time decreases significantly.

Furthermore, information processing systems have increasingly been linked in ever greater computer networks, such as the Internet. The demand for information across these networks is tremendous, and has largely been met by ever increasing the rate in which information has passed between network nodes. For example, fiber optic transmission systems have increased data throughput such that data transmission rates have increased from 1.25 gigabits per second (Gb/s) to 2.5 Gb/s, 10 Gb/s, and are shortly expected to reach rates of 40 Gb/s. While specialized components may be able to receive data at such increased rates, the data rate is often slowed down for processing of the data by less specialized components. A common method of reducing a data rate is to deserialize, or put in parallel, received serial data. For example, serial data transmitted at 40 gigabits per second may be deserialized into a 16 bit bus operating at 2.5 gigahertz. At 2.5 Gb/s, however, skew tolerance for process variations and other factors is often minimal.

## SUMMARY OF THE INVENTION

The present invention provides parallel data de-skew systems and methods. In aspects of the present invention a sample data channel is provided in parallel to parallel data channels to allow for deskewing of the parallel data channels.

The sample data channel carries sample data, which in some aspects is data sampled from data for transmission over or transmitted over the parallel data channels. In some aspects the sample data channel carries a reverse data sample in the opposing direction of transmission as data transmitted over the parallel data channels. In aspects of the present invention the sample data is compared with data transmitted over the parallel data channels to allow for adjustment of skew in signal paths of data transmitted over the parallel data channels.

One aspect of the invention is a de-skew system. The de-skew system comprises a processor configured to receive input data and generate parallel data, a de-skew unit receiving the generated parallel data and a timing signal and adjusting timing of the generated parallel data, based on the timing signal, to generate a plurality of data signals, and a control unit configured to collect portions of the plurality of data signals and to receive a loop data sample and generating the timing signal based on a comparison of the collected portions of the plurality of data signals and the loop data sample.

One aspect of the invention comprises receiving a loop back data sample, determining a data channel specified by the loop back data sample, determining a delay for data from a specific channel when the determined data channel is the specific channel, and delaying data from the specific channel by the determined delay.

One aspect of the invention comprises providing parallel data over a plurality of parallel data lines, successively providing sample data over a sample channel, the sample data corresponding to data of the parallel data, and using the sample data to align the parallel data.

One aspect of the invention is a system including deskew functions comprising an upstream unit providing parallel data to a downstream unit over parallel data channels, a downstream unit receiving the parallel data from the upstream unit over the parallel data channels, and a sample channel coupling the upstream unit and the downstream unit, the sample channel carrying samples of the parallel data.

One aspect of the invention is a system of two units coupled by parallel data lines comprising a first unit providing parallel data over N parallel data lines, a second unit receiving the parallel data over the N parallel data lines, a spare channel in parallel with the N parallel data lines, the first unit providing data of the parallel data lines over the spare channel, the second unit receiving the data of the parallel data lines over the spare channel, a return channel in parallel with the N parallel data lines, the second unit providing data of the parallel data lines over the return channel, the first unit receiving the data of the parallel data lines over the return channel, and at least one unit in the first unit deskewing the N parallel data lines using data of the parallel data lines received over the return channel.

These and other aspects of the present invention will be more readily understood upon review of the accompanying drawings and following detailed description.

## BRIEF DESCRIPTION OF THE DRAWINGS

FIG. 1 is a block diagram of a de-skew system in accordance with aspects of the present invention;

FIG. 2 is a further block diagram of a unit providing a serial/parallel interface function;

FIG. 3 is a flow diagram of a process of providing data on a sample channel;

FIG. 4 is a block diagram of an upstream unit performing a preskew function using data from a sample channel;

FIG. 5 is a block diagram of a downstream unit providing data on a sample channel;

FIG. 6 is a flow diagram of a process performing a de-skew operation with a loopback data sample;

FIG. 7 is a flow diagram of a subprocess for setting the transmit point of a first channel and locating the corresponding loopback data;

FIG. 8 is a flow diagram of a subprocess for determining a delay for the data channel data;

FIG. 9 is a flow diagram of a subprocess for de-skewing the data channels;

FIG. 10 is a semi-schematic diagram of one embodiment of a processor performing a preskew function;

FIG. 11 is a block diagram of an upstream unit performing a preskew function;

FIG. 12 is a semi-schematic diagram of one embodiment of a downstream unit providing a loop-back data sample;

FIG. 13 is a block diagram of one embodiment of a downstream unit providing a loopback sample data;

FIG. 14 is a semi-schematic diagram of portions of a unit performing pre-skew operations;

FIG. 15A is a block diagram of a system utilizing a spare channel and a return channel for deskew purposes;

FIG. 15B is a block diagram of a downstream of a system utilizing a spare channel and a return channel for deskew purposes;

FIG. 15 illustrates a flow diagram of one embodiment of a process for testing and providing data and de-skewing data using a spare channel and a return channel;

FIG. 16 illustrates a schematic of one embodiment of a unit providing a return signal; and

FIG. 17 illustrates a flow diagram of another embodiment of a process for testing and providing data and de-skewing data using a spare channel and a return channel;

FIG. 18 illustrates an exemplary header transmitted with a data sample.

## DETAILED DESCRIPTION

In one aspect of the present invention a sample channel is provided in parallel with a plurality of data channels. The sample channel carries copies of data carried by the data channels, and the sample channel is used to time, or synchronize or de-skew, the data channels.

In one embodiment the sample channel is provided data from the data channels at the receiving side, and the data of the sample channel is received by the transmitting side. The transmitting side compares the data on the sample channel with corresponding delayed data from a data channel to adjust delays in transmission of the data in the data channels. In a further embodiment, a spare channel is used to provide data, from the data channels, from a transmitting side to a receiving side, and a sample channel provides data from the data channels and/or the spare channel from the receiving side to the transmitting side. The use of the spare channel allows for alignment changes of the data in the data channels without potential corruption of data intended for downstream processing.

FIG. 1 illustrates a system in accordance with the present invention. As illustrated, the system includes a serializer/deserializer (SERDES) 101. In alternative embodiments the SERDES is replaced by a processor or other unit, with the processor or other unit containing functions such as, for example, those hereinafter described. The SERDES receives communication data over a transmission link 103. The transmission link is often a fibre optic cable coupled to a photodiode, transimpedance amplifier, and other receiving circuitry (not shown). The SERDES deserializes the communication data and places the deserialized data on a bus 105,including an output bus. In the embodiment illustrated the output bus is a 16-bit bus. The SERDES is therefore a 1:16 deserializer, and the output bus includes 16 data channels.

The output bus is coupled to a processing unit 107. The processing unit processes the deserialized data. As the data transmitted on the output bus includes data received over a plurality of clock cycles from the transmission links, a clock cycle of the output bus is, in the embodiment described, 16 times longer than that of the transmission link. Thus, the processing unit may operate at clock speeds that are a fraction of the clock speed of the SERDES.

The processing unit in various embodiments performs a variety of functions. In one embodiment, the processing unit receives data on the output bus and arranges the data in frames. For example, when the processing unit is utilized as part of a SONET communication system, the processing unit descrambles the data as appropriate and frames the data using, for example, the A1A1A1A2A2A2 framing pattern. In other embodiments the processing unit may perform forward error correction processing or other processing.

The SERDES also receives data from the processing unit over the bus, as illustrated including an input bus. The SERDES serializes the data and provides the data over the transmission link.

Also coupling the SERDES and the processing unit is a sample channel 111. In one embodiment, the processing unit places information from different channels of the output bus onto the sample channel at predefined intervals. In another embodiment the processing unit places information from different channels of the output bus onto the sample channel in response to a command or request from the SERDES. The information provided on the sample channel allows the SERDES to adjust for transmission time variations, or skew, in the data channels.

FIG. 2 further illustrates a further block diagram of a pre-skew system with a SERDES 1400 and a processing unit 1402. The SERDES receives serial data via first transmission connection 1404 and supplies serial data via a second transmission connection 1406. The received serial data is de-serialized and provided as parallel data via a first parallel transmission connection 1408 to the processing unit. A sample or samples of the received parallel data, e.g., a reverse or loop back data sample 1416, is generated and provided back to the SERDES. Based on the received loop back data sample, the SERDES adjusts the timing of the parallel data transmitted to the processing unit.

The processing unit also supplies the SERDES with parallel data via a second parallel transmission connection 1410. The SERDES serializes the received parallel data and transmits the data via the second transmission connection. The SERDES, in one embodiment, also generates and supplies a sample or samples of the received parallel data, e.g., a loop back data sample 1418, back to the processing unit. Based on the received loop back data sample, the processing unit adjusts the timing of the parallel data transmitted to the SERDES. Additionally, the processing unit receives parallel data and sends parallel data to other units (not shown) via a third and fourth transmission connection 1412 and 1414. Accordingly, the SERDES and processing unit have a parallel interface, with the parallel interface including data signals and a sample signal associated with the data signals.

In various embodiments the sample unit selects data streams for transmission to the other component using various criteria. In one embodiment the sample unit periodically selects a particular data stream, with the data streams selected, for example, in round robin fashion. In some embodiments the selected data stream is based on a signal generated by the other component. For example, in one embodiment the other component provides a signal commanding selection of a particular data stream, and in another embodiment the other component provides a signal that serves as a start signal for selection of a particular data stream followed by periodic selection of other data streams.

In various embodiments, the SERDES may be a transmitting unit, or upstream unit, providing data to a downstream unit or a processing unit, such as a framer or FEC processor, may be the transmitting or upstream unit.

FIG. 3 illustrates a flow diagram of a process performed by, for example, the processing unit or SERDES. Those of ordinary skill in the art will recognize that generally the process of FIG. 3, and various other processes, is generally implemented in hardware, with the hardware functions described, for example, through the use of a design language such as HDL, VHDL, or the like. The design language is thereafter synthesized and otherwise processed to provide the hardware component layout.

In Block 201 of the process of FIG. 3 a channel is selected. The channel is one of, for example, 16 data channels. In Block 203 the process creates a header. The header comprises, in one embodiment, a channel number and status word. The channel number is indicative of the selected channel. The status word allows for transmission of further additional information, or other out of data channel information, from the SERDES to the processing unit. The header is described in greater detail in reference to FIG. 18.

In Block 205 the process copies data from the selected channel. The copied data is a predetermined number of bytes from the selected channel. In Block 207 the process transmits the header and copied data from the processing unit to the SERDES. The header and copied data together comprise a reverse data sample. In Block 209 the process determines if an exit has been commanded. If no exit has been commanded, the process returns to Block 201 and selects a further channel for transmitting a further data sample. In one embodiment the channels are selected on a round-robin basis, although in other embodiments other selection criteria are used.

FIG. 4 illustrates a block diagram of an upstream unit in accordance with aspects of the present invention. In FIG. 4, a serial data stream 1402 is provided to a processor 1404. The processor in this case may perform a deserialization function, or other functions, and provides a parallel data stream 1406. The parallel data stream is provided to a pre-skew unit 1408. The pre-skew unit delays each of the data streams making up the parallel data stream on a data stream by data stream basis. The pre-skew unit provides a parallel output data stream 1410, which is provided to downstream units

The parallel output data stream is also provided to a control 1412. The control also receives a serial sample channel data stream 1414. The serial sample channel data stream is provided, for example, from a downstream receiving unit. The serial sample data channel stream includes data selected from the parallel output data stream. On a varying basis, at any particular moment the serial sample channel data stream contains data from a particular one of the parallel output data streams. The control provides a signal to a pre-skew unit which indicates on a channel by channel basis the appropriate delay per channel.

In operation, the control compares the data in the serial sample channel data stream with corresponding data in the parallel output data stream in order to set a pre-skew delay for that particular data stream in the parallel output data stream. In one embodiment the control includes delay elements so as to delay the data stream from the parallel output data stream to allow for comparison with corresponding data from the serial sample channel data stream, which is a looped back signal from the other unit. In another embodiment the parallel output data stream is provided a repeating pattern which allows for the determination of the delay.

FIG. 5 illustrates a downstream unit in accordance with the aspects of the present invention. The downstream unit receives a parallel data stream 1502. The parallel data stream is provided to a processor 1504 which processes the data and provides a further output data stream 1506. The processor, for example, may perform a framing function in one embodiment or a forward error correction function in another embodiment. The parallel data stream is also provided to a sample unit 1508. The sample unit selects one of the data channels forming the parallel data stream and provides the data in the selected data channel to the originating unit in a sample channel 1510.

The upstream unit performs a process such as illustrated in the flow diagram of FIG. 6. In the process of FIG. 6 a first data channel is set to a transmit point in Block 901 and corresponding loop back data is located. In Block 903 a delay for the first data channel is determined so as to match the loop back data. In Block 905 the transmit point of each of the remaining channels is set so as to match data delayed by the delay with the loop back data for each channel.

A flow diagram of a subprocess for setting the transmit point of the first data channel and locating the corresponding loop back data is illustrated in FIG. 7. In Block 1001, a deskew circuit for the first data channel is set at its midpoint. In Block 1003 data from the first channel is selected to be provided to a delay element. In Block 1005 the process receives a reverse or loop back data sample on a reverse or loop back channel. In Block 1007 the process determines the corresponding data channel identified by the loop back data sample. In Block 1009 the process determines if the corresponding data channel for the loop back data sample is the first data channel. If the corresponding data channel is not the first data channel the process returns to Block 1003. Otherwise the process returns.

FIG. 8 illustrates a flow diagram of a subprocess for determining a delay for the data channel data. In Block 1101 the process compares data from the delay element with the data from the loop back data sample. In Block 1103 the process determines if the data matches. If the data does not match, the process adjusts the delay time of the delay element in Block 1105 and returns to Block 1101. If the data matches, the process returns.

FIG. 9 illustrates a flow diagram of a subprocess for de-skewing the remaining data channels. In Block 1201 the process selects a data channel to provide data to the delay element. In Block 1203 the process receives loop back sample data. In Block 1205 the process determines if the loop back sample data corresponds to the selected channel. If the loop back sample data does not correspond to the selected channel, the process returns to Block 1203. If the loop back sample data corresponds to the selected channel, the process compares the data in the loop back sample data with the delayed data from the selected channel in Block 1207. If the data matches, the process returns to Block 1201 and selects another channel. If the data does not match in one embodiment the process determines a skew delay for the data channel based on the results of the comparison. In the embodiment illustrated in FIG. 10 the process adjusts the skew of the selected data channel in Block 1209, and thereafter returns to Block 1203 to await receipt of a further loop back data sample for the selected channel.

A semi-schematic diagram of one embodiment of a unit performing a pre-skew function is illustrated in FIG. 10. The unit includes a processor, which in a SERDES, for example, may perform deserialization. The processor provides parallel data to de-skew, or pre-skew, units 63a-p. Outputs of the de-skew units are provided to output drivers 65a-p for transmission to a processing unit (not shown). Outputs of the de-skew units are also provided to multiplexers 67a-e for selection of data from data channels DATA0-DATA15 by a controller 161. The selected data from multiplexers 67a-d are provided to multiplexer 67e which provides one of selected data to a delay element 69. In one embodiment the delay element is a string of latches, as also may be the case in some embodiments previously described. The delay element contains sufficient delay to approximate the round trip time of data from, for example, a SERDES and returned on the loop back channel L2. The delay element is controlled or adjusted by the de-skew controller.

In a further embodiment, both the upstream unit and the downstream unit include sample generator and de-skew circuitry, and multiple sample/control channels, e.g., data sample channel S6 via buffer 65g, are provided between the units. Thus, in various embodiments, a control signal is provided to the units so that one unit performs de-skewing while the other unit performs error monitoring, and also provides additional control signals between the units.

In FIG. 11, a processor 51 receives input data I5. The input data, in one embodiment, is information provided via an optical link. More commonly, the information is provided by a parallel interface, thereby reducing processing efforts by higher speed SERDES units. The processing unit de-serializes the data and supplies the data, in parallel form, to a deskew unit 53. Based on a timing signal T5, the deskew unit forwards data to drivers 55. The drivers transmit the data to, in one embodiment, another processing unit (not shown).

The deskew unit also forwards data to a selection unit 57. A control unit 59 is coupled to the selection unit and commands the selection unit to generate or select specific samples or portions of data received from the deskew unit. In one embodiment, the data selection is performed in a round-robin fashion. The selection unit aggregates the samples of data to form a data sample signal S5. The selection unit transmits the data sample signal to the drivers 55 and a delay unit 151. The delay unit is adjustable by the control unit and adjusts the timing of the data sample signal. The delayed sample signal is then supplied to the control unit.

The control unit also receives a loop data sample L1. The control unit compares the loop data sample to the delayed data sample. Based on the comparison of the loop data sample and the delayed data sample, the control unit generates the timing signal. As such, the control unit is able to adjust the data from the deskew unit to be synchronized with the timing of the loop data sample.

FIG. 12 illustrates a semi-schematic diagram of one embodiment of a downstream unit. In FIG. 12, a data processor 95 receives signals on data channels DATA0-DATA15. The signals are provided to buffers 91a-p, and then to latches 93a-p. The output of the latches is provided both to a data processor 95 and multiplexers 97a-d under control of a sample controller 99. The five 4x1 multiplexers 97a-97e are utilized to select a loop back data sample. Accordingly, data samples DATA0-DATA3 are provided to a first multiplexer 97a, data samples DATA4-DATA7 are provided to a second multiplexer 97b, data samples DATA8-DATA11 are provided to a third multiplexer 97c, and data samples DATA12-DATA15 are provided to a fourth multiplexer 97d. The output of each of the first, second, third, and fourth multiplexers are in turn provided to a fifth multiplexer 97e, whose output is provided to a sample controller 99.

The sample controller controls the selectors of the multiplexers 97a-e so as to be able to select data from a particular data channel. The sample controller also provides data to the output buffer for the data sample. Thus, the sample controller selects a data channel, sets the appropriate selectors from the multiplexers, receives the output of the fifth multiplexer, and appends the data from the data channel to the header to form a loop back data sample L7. The sample controller provides the forward data sample to the buffer 91g for transmission to another unit, for example, a SERDES (not shown).

In the alternative embodiment the processing unit performs functions analogous to the functions described with respect to the process of FIG. 4. Accordingly, and as illustrated in FIG. 13, input drivers 172 receive input data I7. The input data, in one embodiment, is information provided from a SERDES (not shown). The input data is in parallel form and, in one embodiment, includes sixteen data channels. The input drivers supply the parallel data to a buffer unit 173.

A clock signal C7 is also supplied to the buffer unit. Based on the clock signal, the buffer unit transmits the data to a data processor 175. The buffer unit also forwards data to a selection unit 177. A control unit 179 is coupled to the selection unit and commands the selection unit to generate or select specific samples or portions of data received from the buffer unit. In one embodiment, the data selection is performed in a round-robin fashion. The control unit aggregates the samples of data to form a data sample signal. The data sample signal is supplied to output drivers 180 which, in one embodiment, transmits the data sample S7 to a SERDES (not shown).

Thus, for example, the SERDES and the processing unit, both previously described, are able to de-skew parallel data channels, and to do so without the use of training patterns or the like being transmitted in the data channel, although in some embodiments it is convenient to use such patterns to further data channel de-skew. In one embodiment, for example, pre-emphasis adjustment to the outgoing waveform is adjusted by SERDES, i.e., a downstream unit, to increase the ability of the processing unit to read transmitted data. In a further embodiment a control signal from the processing unit to the SERDES provides information to the SERDES for use in waveform shaping. In some embodiments the information provides information regarding the nature or characteristics of the processing unit, and in other embodiments the information provides commands as to waveform adjustment. In addition, in further embodiments the system continues operation once de-skew is complete in order to monitor alignment of the data in the data channels, and to recommence de-skew operations when data is out of alignment. Further, in various embodiments the sample channel is also used to provide information on the status of the interface or other out of channel information.

The transmitting, or upstream, device adjusts the transmit time of the data channels to perform de-skewing. Thus, the receiving, or downstream device, provides a loop back data sample on a sample channel to the transmitting device. The processing unit may be the downstream unit, although in alternative embodiments each may be either or both the downstream unit and the upstream unit, as would be understood by those of ordinary skill in the art.

FIG. 14 illustrates portions of a preskew function. Each of 16 parallel data lines are provided to buffers 1507. The buffers may take the form, for example of FIFOs, including tapped FIFOs. Each of the FIFOs receives potentially varying control or clocking signals, allowing for alignment and synchronization of the parallel data lines.

Referring somewhat more specifically to the embodiment of FIG. 14, in FIG. 14 an external reference clock 1501 is supplied to a phase-locked loop (PLL) or digital locked loop (DLL) 1505. The PLL or DLL is coupled to a clock phase generator 1503. The phase-locked loop circuit ensures that the clock signal supplied to the clock phase generator does not vary, e.g., remains in phase, from the external reference clock. In one embodiment, the external reference clock 1501 is supplied directly to a clock phase generator 1503.

The clock phase generator slices the reference clock. In the embodiment described, the phase generator creates a number of clock signals, each of the same frequency, but phase-shifted with respect to each other. Thus, in one embodiment the clock phase generator creates the external reference clock into sixteen clock signals, each phase shifted 22.50 degrees from another clock signal.

The clock phase generator supplies an output signal to each FIFO unit 1507a-1507q. Each output signal is based on the sixteen clock signals. In one embodiment the output signal is merely one of the sixteen clock signals. In another embodiment the output signal is a weighted sum of the sixteen clock signals, in one instance as is described in U.S. Patent Application No. 09/265,725, the disclosure of which is hereby incorporated by reference. A phase select per channel signal 1509 selects, or determines, the clock signal, or weights of clock signals, for the output signal supplied to each FIFO unit for each channel. In other words, the phase select per channel signal 1509 is supplied to the clock phase generator to select the output from the clock phase generator, e.g., a clock signal from the clock phase generator. The phase select per channel signal is provided by a de-skew controller (not shown in FIG. 14).

The FIFO units also receive a clock signal 1511 and a respective data channel DATA0 - DATA15, as inputs. In one embodiment the clock signal has a substantially higher rate than the data rate for the data channels. Information, digital data, on each data channel is clocked into the corresponding FIFO unit by the clock signal. Information on each data channel is then clocked out by the output from the clock phase generator as selected by the phase select per channel signal.

Thus, this allows digital data for each channel to be delayed in a FIFO unit or register for that channel, and released from the register at a time that is a controlled fraction of a clock period. The clock period, which is externally controlled, is not subject to variations due to manufacturing process or environmental variables. Therefore, the de-skew time, the time differences between channels, is referenced to the externally controlled clock. In many instances, it would be desirable for the de-skew time to remain constant. However, there may be cases where unavoidable delays in the data transmission path vary in a known way with environmental variables such as temperature. In this case, the fraction of the clock period used to de-skew the channel could be adjusted to compensate the known dependency in the unavoidable delay. As such, the need for clock recovery on the downstream side of the interface could be reduced.

FIG. 15A is a block diagram of two units, with a first unit providing parallel data streams to a second unit. The first unit also provides the second unit a spare signal. The second unit provides the first unit a return signal. The spare signal includes data corresponding to data on various data streams forming the parallel data streams. The return signal includes data provided on the parallel data streams and the spare signal.

As illustrated, the first unit is a processing unit 2500 and the second unit is a SERDES 2502. Accordingly, the SERDES serializes data on parallel data streams 2508 and provides serialized data 2514 over a serial transmission link 2514. A spare data stream 2510 is provided by the processing unit to the SERDES. The SERDES provides a return data stream 2512 to the processing unit.

In operation, data is provided over N of the N+1 lines providing data to the SERDES, with one of the N+1 lines being potentially deskewed at any given instant. Deskew is accomplished by providing data from the line being deskewed back to the processing unit as the return signal, with the processing unit determining preskew for the line. If the line being deskewed provides data meant for further downstream users, the spare channel may be provided data normally provided by the line being deskewed to allow for passage of the data without temporary potential corruption due to the deskewing of the data line.

FIG. 15B is a block diagram of an embodiment of a second unit of FIG. 15A. The second unit receives N+1 parallel signals, including N parallel signals 1202 and a spare signal 1200. The parallel signals are provided to a sample unit 1208. The sample unit selects one of the N+1 parallel signals and outputs the selected signal as a return signal 1210. The sample unit also provides N parallel data signals 1212 to a processor 1204. As illustrated in FIG. 15B, the processor outputs a serial signal 1206, and therefore performs a serialization function.

FIG. 15 is a flow diagram of an embodiment of a process for performing deskew using a spare channel and a return channel. In Block 253 data is copied from a data channel onto a spare channel. The data channel is one of a plurality of data channels provided by a first unit to a second unit. The spare channel is also provided by the first unit to the second unit. The second unit returns the data on the spare channel to the first unit on a return channel in Block 253. The spare channel is then deskewed by the first unit, using for example preskew techniques earlier discussed.

The process then loops through each of the data channels, deskewing each in turn, although in one embodiment the spare channel is deskewed after deskewing of each data channel. In somewhat more detail, in Block 257 the process selects a data channel. In Block 261 the process copies data from the selected data channel to the spare channel. In Block 263 the process returns, from the second unit to the first unit, the data from the selected data channel on the return channel. The process also acts on the copied data on the spare channel as if it were provided on the data channel. This further ensures use of data on the selected data channel during deskew adjustments to the data channel. In Block 265 the process deskews the selected data channel as earlier discussed.

In Block 267 the process determines if an exit has been commanded. If so the process ends. If no exit has been commanded the process determines if all data channels have been selected in Block 269. If not all data channels have been selected the process selects another data channel in Block 257 and continues. If all data channels have been selected the process returns to Block 251 and repeats.

FIG. 16 is a gate level diagram of one embodiment of a circuit providing data channels and a sample channel. For readability, only four data channels DATA0-DATA3 are shown, but, as one skilled in the art would recognize, numerous data channels with associated logic gates could be added in accordance with the illustrated embodiment. Latches 201a-201d receive data from data channels DATA0-DATA3. In the embodiment shown, the data channels provide differential inputs. The embodiment illustrated also provides for a spare channel input, and latch 201e receives data from a spare data channel SPARE0. As with the other data channels, the spare data channel provides differential inputs. A differential clock signal is also supplied to each of the latches 201a-201e via buffer 223.

The outputs of latches 201a-201d are coupled to respective 2x1 differential multiplexers 209a-d. As such, in one embodiment, at a rising or falling edge of the clock signal, the buffers are triggered. Thus, the data on the data channels DATA0-DATA3 are supplied to respective multiplexers 209a-d from latches 201a-d. Latch 201e is also coupled to each multiplexer 209a-d. Thus, at the rising or falling edge of the clock signal, the data on the spare data channel is also provided to the multiplexers 209a-d.

The multiplexers are coupled to latches 211a-d. Selection signals are provided to the multiplexers 209a-d. Based on the selection signals supplied to each multiplexer, a set of inputs are chosen and supplied to the respective latches 211a-d. The latches 211a-d are also supplied the differential clock signal via buffer 223. At the rising or falling edge of the clock signal, the selected inputs from the respective multiplexers are clocked out of the buffers as outputs for the selection unit. Thus, the multiplexers 209a-d allow for selection of the corresponding data channel or the spare channel based on the selection signals.

Buffers 201a-201b are also coupled to a first multiplexer 203a. Similarly, latches 201c-d are coupled to a second multiplexer 203b. The first and second multiplexers are respectively coupled to latches 205a-205b. Selection signals are provided to both multiplexers. Based on the selection signals, a particular input or set of inputs are chosen. The selected input or inputs are supplied to the respective latches 205a-205b. The latches 205a-205b also receive the differential clock signal. In one embodiment, at a rising edge of the clock signal the buffers output the selected inputs to a third multiplexer 207a.

Similar to the other multiplexers, the third multiplexer receives selection signals and is coupled to a latch, latch 213a. As such, based on the selection signals, a set of inputs are selected and supplied to the latch 213a. The latch 213a also receives the differential clock signal and, in one embodiment, at a rising edge of the clock signal, the latch outputs the selected inputs to a multiplexer 209e. Thus, in operation the multiplexers 203a, 203b, and 207a serve as a 4x1 multiplexer, albeit with a delay. The effective 4x1 multiplexer selects one of the four data inputs and places it on a path allowing for placement on the return, or sample, channel.

The multiplexer 209e also receives data from the spare data channel via latches 201e, 205c and 213b. Each latch also receives the differential clock signal from buffer 223. Accordingly, at the rising or falling edge of the clock signal, the data on the sample data channel is also provided to each latch in succession, latch 201e to latch 205c to latch 213b, and finally to the multiplexer 209e.

The multiplexer 209e is also coupled to a latch 211e. Selection signals are provided to the multiplexer 209e. As such, based the selection signals, a set of inputs, e.g., from latch 213a, the data from data channels DATA0-DATA3, or from latch 213b, the data from spare data channel SPARE0, are chosen and supplied to the latch 211e. The latch 211e is also supplied the clock signal. At the rising or falling edge of the clock signal, via the latch 211e, the selected inputs from the multiplexer, is supplied to another multiplexer 215a. The multiplexer 215a also receives input from a latch 211f. The latch 211f is coupled to outputs of an XOR gate 221. The XOR gate compares (exclusive ORs) the outputs from the latches 213a and 213b, data from the spare data channel and the data from one of the data channels DATA0-DATA3. The XOR gate thereby simplifies spare channel to data channel comparison. The latch 211f also receives the differential clock signal and thus at the rising or falling edge of the clock signal, the latch 211f provides the output from the XOR gate to the multiplexer 215a.

The multiplexer 215a also receives selection signals. In one embodiment, the selection signals are supplied by a controller (not shown). Based on the selection signals, a set of inputs from the latches 211e or 211f are selected and supplied as an output of the selection unit. Accordingly, input data from the data channels DATA0-DATA3, from the spare data channel SPARE0, or the comparison of the data from the data channels to the data from the sample data channel, is supplied as an output from the selection unit.

The selection signals supplied to the multiplexers 203a,b are generated by a series of latches 217a-c coupled together. Similarly, the selection signals supplied to the multiplexer 207a are supplied by latch 217b. The first latch 217a, in one embodiment, receives increment signals from a controller (not shown). A set of the first latch's outputs is fed back as inputs to the latch. As such, the first latch acts as a counter and is triggered by the increment signals. Another set of the first latch's outputs are also supplied to the second latch 217b, the third latch 217c and inputs to OR gates 219a-d. Furthermore, the set of first latch's outputs are also supplied to the multiplexer 209e, as selection signals. Outputs from a corresponding set of OR gates 219e-h also supply inputs to the respective OR gates 219a-d. The outcome or outputs of the OR gates 219a-d are the selection signals to multiplexers 209a-d, respectively.

The outputs of the OR gates 219e-h depend on outputs from the latches 217b and 217c. The output from the latch 217c is also fed back as an input to latch 217b. Thus, the latches 217a-c act as counters and are triggered or count up based the increment signals supplied to latch 217a. Therefore, the latch and OR gates provide control to select data, data samples or a comparison of data samples by generating the appropriate selection signals. As such, input data from the data channels DATA0-DATA3, the spare data channel SPARE0 and a comparison of the data from the data channels DATA0-DATA3 and the sample data channel, outputs from the selection unit via a series of multiplexers and buffers based on the timing of the edges of the clock signal and the value of the selection signals.

In one embodiment, and as illustrated in FIG. 16, the selection signals are based on the increment signal. The increment signal is provided to latch 217a, and latches 217 and XOR gates 219 act on the increment signal and/or resulting signals to form the selection signals. In one embodiment the increment signal is provided by a controller, or a sample unit, in the upstream device. Beneficially, in one embodiment a signal is returned to the upstream device, with the signal generated by the XOR gates, so as to inform the upstream device as to the status of the selection signals and information related thereto.

In one embodiment, the data on data channel DATA0 is copied on to the return channel RDATA. Specifically, data on the data channel DATA0 is supplied to multiplexer 215a via buffers 201a, 205a, 213a and 211e, and multiplexers 203a, 207a and 209e for output at the return channel RDATA. The selection signals to each of the respective multiplexers 203a, 207a and 209e are logic zeros. As such, the first set of inputs, i.e., data from data channel DATA0, are selected. At each rising or falling clock edge the data is clocked through the buffers 201a, 205a, 213a and 211e. Thus, in the embodiment shown, in four clock cycles, the data from data channel DATA0 is supplied to the inputs of multiplexer 215a.

The selection signal coupled to multiplexer 215a is also logic zero and thus the data is transferred to the return channel RDATA and then returned. In one embodiment, the data is returned to a processing unit (not shown). In one embodiment the processing unit receives the returned data and de-skews the data. The de-skewed data is then transmitted which is received by the downstream device via the input spare data channel.

Thus, the data present on data channel DATA0 may again be copied on to the return channel RDATA. The input spare channel SPARE is selected and output from the output data channel OUT0. Specifically, data from the spare channel SPARE is received and provided to the buffer 211a, via buffer 201e and multiplexer 209a. The increment signal coupled to buffer 217a is activated which causes the buffer to count up. Thus, a logic one is provided to each of the OR gates 219a, b, c, d which in turn provides a logic one to respective multiplexers 209a-e. As such, data from the input spare data channel is selected and supplied to buffer 211a. Within two clock periods the data is output from the output data channel OUT0. The data copied on the output spare channel is returned and the data is de-skewed.

The data from the data channel DATA0 is then selected and output from the output data channel OUT0. Specifically, data from the data channel DATA0 is received in buffer 201a and provided to the buffer 211a, via multiplexer 209a. The selection signal provided to the multiplexer 209a remains a logic one and thus the data from the data channel DATA0 is selected and passed to the buffer 211a. In about two clock cycles, the data is output via output data channel OUT0.

The data from data channel DATA1 is copied on to the return channel RDATA. Specifically, data on the data channel DATA1 is supplied to multiplexer 215a via buffers 201c, 205b, 213a and 211e, and multiplexers 203b, 207a and 209e for output at the return channel RDATA. The selection signals to each of the respective multiplexers 203b, 207a and 209e are logic zeros. As such, the first set of inputs, i.e., data from data channel DATA1, are selected. At each rising or falling clock edge the data is clocked through the buffers 201c, 205b, 213a and 211e. Thus, in the embodiment shown, in four clock cycles, the data from data channel DATA1 is supplied to the inputs of multiplexer 215a.

The selection signal coupled to multiplexer 215a is also logic zero and thus the data is transferred to the return channel RDATA and then returned. In one embodiment, the data is returned to a processing unit (not shown). In one embodiment the processing unit receives the returned data and de-skews the data. The de-skewed data is then transmitted and, in one embodiment, received by the selection unit via the input spare data channel.

If another data channel, e.g., data channel DATA1, is to be selected the process repeats. For instance, assume data from data channel DATA1 is selected next. The counter of buffers 217a-c, are incremented and the selection signal provided to multiplexer 209c, selects the data from data channel DATA1.

Data from data channel DATA3 is copied onto the return channel and returned. The counter is incremented again, such that the selection signal provided to multiplexers 203b, 207a and 209e select the data from the data channel DATA3 to be provided to the output spare data channel RDATA. The returned data is then de-skewed.

Data channel DATA3 is then selected. The counter of buffers 217a-c, are incremented and the selection signal provided to multiplexer 209d, selects the data from data channel DATA3. Thus, data is output via output data channel OUT3.

Data from data channel DATA2 is copied onto the return channel. For example, the counter is incremented, such that the selection signal provided to multiplexers 203a, 207a and 209e select the data from the data channel DATA2 to be provided to the return data channel RDATA. The returned data is then de-skewed.

Next, data channel DATA2 is selected. The counter of buffers 217a-c, are incremented and the selection signal provided to multiplexer 209b, selects the data from data channel DATA2. Thus, data is output via output data channel OUT2.

Data from data channel DATA0 is copied onto the return channel and returned. The returned data is then de-skewed. The process continuously repeats, such that all the data channels are selected, returned and de-skewed. If all the data channels have been de-skewed, in one embodiment, the process ends.

FIG. 17 illustrates a flow diagram of another embodiment of de-skewing data channels. In block 2001, a test data channel supplies data to the spare data channel SPARE. The data is output via the return channel RDATA, in block 2003. The data from the return channel RDATA is then de-skewed by observing the returned data, in block 2005. In block 2007, a predetermined test data is provided to all data channels, i.e., data channels DATA0-DATA3. In block 2009, one of the data channels is selected and the received test data is XOR'ed to the spare data channel by XOR gate 221, in block 2011. The output of the XOR gate is returned via buffer 211f and multiplexer 215a, in block 2013. The output of the XOR gate is then observed to determine if the selected data channel should be de-skewed, in block 2015. If so, the selected data channel is de-skewed, in block 2017. Otherwise or once the selected data channel is de-skewed, the process is repeated by continuing to block 2009, such that all the data channels are selected and de-skewed. As such, each data channel is selected in turn and the received test data is XOR'ed to the spare data channel and the output observed for de-skew. In one embodiment, the process continuously repeats.

In one embodiment, as illustrated in FIG. 18, the data sample frame includes a header containing a fixed pattern, the replicated channel number, the sample length and a status word. The remainder of the frame includes a fixed quantity of data sample from the selected data channel. In one embodiment, where interfaces use asymmetric de-skew hardware replacement, the header is not utilized. As such, one extra signal, the clock signal, is supplied to indicate the start of the data sample switching to the first channel. The period of the clock would be the number of channels times the sample size.

Referring again to FIG. 18, the header includes a thirty-two bit frame delimiter containing the pattern of A1A1A2A2. Following the delimiter, eight bits are provided to indicate the channel number CH, i.e., the channel from which the sample data will be taken. The sample length SL is indicated by an eight bit word. In one embodiment, the length of a sample is indicated in thirty-two bit words. Finally, a sixteen-bit status word ST0 and ST1 is provided to indicate, for example, errors. Accordingly, the total header size is sixty-four bits in the illustrated embodiment. A sixty-four bit data sample D00-D03 and D10-D13 follows the header.

Thus, the present invention provides reverse data de-skew methods and systems. Although described in certain specific embodiments, it should be understood that the present inventions may be practiced otherwise than as specifically described, the bounds of the present invention being set by claims and their equivalents supported by the description herein.